

# Best Practices for Storing the Product Library Workspace in an Enterprise Geodatabase for Oracle

An Esri® White Paper  
October 2012



Copyright © 2012 Esri  
All rights reserved.  
Printed in the United States of America.

The information contained in this document is the exclusive property of Esri. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by Esri. All requests should be sent to Attention: Contracts and Legal Services Manager, Esri, 380 New York Street, Redlands, CA 92373-8100 USA.

The information contained in this document is subject to change without notice.

Esri, the Esri globe logo, ArcSDE, ArcCatalog, ArcMap, ArcGIS, SDE, Spatial Database Engine, [esri.com](http://esri.com), and @esri.com are trademarks, service marks, or registered marks of Esri in the United States, the European Community, or certain other jurisdictions. Other companies and products or services mentioned herein may be trademarks, service marks, or registered marks of their respective mark owners.

# Best Practices for Storing the Product Library Workspace in an Enterprise Geodatabase for Oracle

## An Esri White Paper

| <b>Contents</b>                                              | <b>Page</b> |
|--------------------------------------------------------------|-------------|
| Introduction.....                                            | 1           |
| Overview of ArcSDE DBTUNE.....                               | 1           |
| Disk Configuration.....                                      | 2           |
| Reducing Disk I/O Contention.....                            | 2           |
| Transparent Data Encryption .....                            | 2           |
| Step 1: Create Data Files.....                               | 4           |
| Step 2: Create the PRODLIB User .....                        | 5           |
| Step 3: Modify DBTUNE .....                                  | 6           |
| Step 4: Create the Product Library Database Connection ..... | 6           |
| Step 5: Configure Oracle Parameters.....                     | 7           |
| Step 6: Configure ArcSDE Parameters.....                     | 7           |
| Step 7: Create the CKB_USERS Role.....                       | 7           |
| Step 8: Create the Product Library Workspace.....            | 7           |
| Step 9: Verify the Storage.....                              | 9           |
| Step 10: Register as Versioned .....                         | 9           |

| <b>Contents</b>                                                         | <b>Page</b> |
|-------------------------------------------------------------------------|-------------|
| Step 11: Validate Permissions and Roles.....                            | 10          |
| Grant Permissions Using ArcCatalog.....                                 | 10          |
| Step 12: Configure Log File Tables.....                                 | 11          |
| Create Log File Tablespaces.....                                        | 11          |
| Change DBTUNE Log File Parameters.....                                  | 12          |
| Create Log File Tables.....                                             | 12          |
| Step 13: Create the ArcSDE Product Library User .....                   | 12          |
| Step 14: Create Database Connections for Product Library<br>Users ..... | 13          |
| Step 15: Assign Product Library Permissions .....                       | 13          |
| Step 16: Add New Users to the Product Library .....                     | 13          |
| Assigning Permissions to Users.....                                     | 15          |
| Replication .....                                                       | 17          |
| Conclusion .....                                                        | 17          |

# Best Practices for Storing the Product Library Workspace in an Enterprise Geodatabase for Oracle

**Introduction** The product library is part of the Esri® Production Mapping extension. It is a geodatabase that allows multiuser environments to centralize information and behavior for cartographic and digital data production. Production business rules, documents, and spatial information are stored inside the product library, allowing an organization to enforce and standardize production. Data model information, data validation rules, geographic extents, symbology rules, and map documents can all be managed inside the product library as examples of production business rules. In other words, the product library is essentially a geographic document management system. When stored in an enterprise geodatabase, the workspace supports versioning. This white paper is intended to help database administrators establish the product library workspace in an enterprise geodatabase for Oracle. The enterprise geodatabase uses ArcSDE® technology as the gateway between geographic information system (GIS) clients and Oracle.

## Overview of ArcSDE DBTUNE

DBTUNE storage parameters let you control how ArcSDE technology creates objects within an Oracle database. They allow you to determine things such as how to allocate space to a table or index and which tablespace a table or index is to be created in, as well as other Oracle-specific storage attributes. They also let you specify one of the available storage formats for the geometry of a spatial column.

The DBTUNE storage parameters are stored in the DBTUNE table. The DBTUNE table, along with all other metadata tables, is created during the setup phase that follows the installation of ArcSDE. ArcSDE installation creates a dbtune file under the etc directory from which the DBTUNE table is populated. If no dbtune file is present during setup, the DBTUNE table will be populated with default values.

When a large number of database connections access the same files in the same location on disk, database performance is slower, because the connections are competing with one another for the same resources. To reduce this competition, you can store database files in different locations on the disk.

Thus, DBTUNE can be modified to store the product library tables in separate data files in different locations on the disk. This will reduce disk contention and improve database input/output (I/O).

Standard GIS storage recommendations favor keeping index and log files separate from vector and tabular business tables. For performance reasons, it is better to position the business, feature, and spatial index tables separately and position tablespace data files

based on their usage pattern. For a multiversioned, highly active editing geodatabase, database files of the VERSIONS tablespace may be separated and dispersed across available disks to avoid I/O contention.

### ***Disk Configuration***

Large production enterprise geodatabase systems should employ a hardware striping solution. The best strategies for disk and data organization involve spreading your data across multiple disks so that more spindles actively search for it. This can increase disk read time and decrease disk contention. However, too many disks can slow down a query. There are two main ways of achieving striping: tablespaces and redundant array of independent disks (RAID). You can also combine the two by creating tablespaces within disk arrays. You can employ data segregation strategies; keeping tables from indexes or certain types of tables from other tables will improve performance and alleviate administrative burdens.

Suggested Oracle optimal configuration is as follows:

- n Disk 0—Oracle/Application Software
- n Disk 1—SYSTEM, Control File 1
- n Disk 2—RBS, TEMP, Control File 2
- n Disk 3—REDO 1, 2, 3, Export Files
- n Disk 4—Feature Data Tables
- n Disk 5—Spatial Index Data Tables
- n Disk 6—Attribute Data/Business Tables
- n Disk 7—Oracle Indexes

### ***Reducing Disk I/O Contention***

As a rule, you should create the largest possible database files, based on the maximum amount of data you estimate the database will contain, to accommodate future growth. By creating large files, you can avoid file fragmentation and get better database performance. In many cases, you can let data files grow automatically; just be sure to limit autoextend by specifying a maximum growth size that leaves some hard disk space available. By putting different tablespaces on different disks, you can also minimize physical fragmentation of your files as they grow.

Below is a suggested design to reduce disk I/O contention:

| File Type            | Database Activity | Move File to Disk With                   |
|----------------------|-------------------|------------------------------------------|
| Redo log             | Frequent edits    | Relatively low I/O                       |
| Redo log             | Few or no edits   | Moderate I/O                             |
| Undo log files       | Frequent edits    | Low I/O but separate from redo log files |
| System data          | Frequent edits    | Moderate I/O                             |
| Temporary tablespace | Few edits         | High I/O                                 |

### ***Transparent Data Encryption***

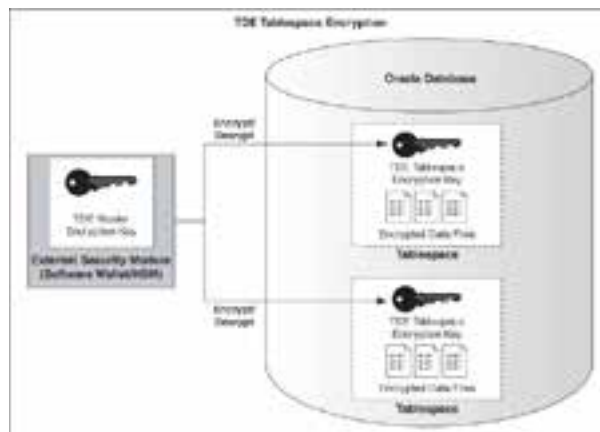
Transparent data encryption (TDE) enables you to encrypt sensitive data, such as credit card numbers, stored in tables and tablespaces. Encrypted data is transparently decrypted for a database user or application that has access to data. TDE helps protect data stored on media in the event that the storage media or data file is stolen. Oracle Database uses authentication, authorization, and auditing mechanisms to secure data in the database but not in the operating system data files where data is stored.

J10023

To protect these data files, Oracle Database provides TDE. TDE encrypts sensitive data stored in data files. To prevent unauthorized decryption, TDE stores the encryption keys in a security module outside the database.

Benefits of using TDE include the following:

- n As a security administrator, you can be sure that sensitive data is safe in case the storage media or data file is stolen.
- n Implementing TDE helps you address security-related regulatory compliance issues.
- n You do not need to create triggers or views to decrypt data for the authorized user or application. Data from tables is transparently decrypted for the database user and application.
- n Database users and applications need not be aware of the fact that the data they are accessing is stored in encrypted form because data is transparently decrypted.
- n Applications need not be modified to handle encrypted data. Data encryption and decryption are managed by the database.
- n Key management operations are automated, so the user or application does not need to manage encryption keys.



*TDE Tablespace Encryption*

See Oracle documentation on how to configure TDE tablespace encryption:

[http://download.oracle.com/docs/cd/E11882\\_01/network.112/e10746/asotrans.htm#ASOAG9579](http://download.oracle.com/docs/cd/E11882_01/network.112/e10746/asotrans.htm#ASOAG9579)

```
-----*
--Configure TDE Oracle Enterprise Manager - OEM
-----*/

--Create the wallet folder
mkdir C:\oracle\admin\wallets
OEM > login as sys / sysdba
OEM > Server > Transparent Data Encryption
```

```
Advanced Options > Change Location

Host Credentials
Username: avworld\dba_ora
Password: xxxxxxxx

Configuration Method: File System

Encryption Wallet Directory: C:\oracle\admin\wallets

OK

Create Wallet > Local Auto-Open Wallet > Create

Host Credentials
Username: avworld\dba_ora
Password: xxxxxxxx

Wallet Password: walletadmin

Continue

--Backup the wallet folder
cd C:\oracle\admin
zip -r wallets wallets

/*-----
--Configure TDE Manually
-----*/
--Create the wallet folder

mkdir C:\oracle\admin\wallets

--Add wallet location to sqlnet.ora

ENCRYPTION_WALLET_LOCATION =
(SOURCE =
(METHOD = FILE)
(METHOD_DATA =
(DIRECTORY = C:\oracle\admin\wallets\%ORACLE_SID)
)
)

Note: The default encryption wallet location is $ORACLE_BASE/admin/<global_db_name>/wallet.
If you want to let Oracle manage a wallet in the default location then there is no need to
set the ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora.

--Generate a master key

alter system set encryption key identified by "walletadmin";

--See the status of the wallet

select * from v$encryption_wallet;

--Make the wallet auto login

set ORACLE_SID=prodlibdb

orapki wallet create -wallet C:\oracle\admin\wallets -auto_login -pwd walletadmin

--Backup the wallet folder

cd C:\oracle\admin
zip -r wallets wallets
```

## Step 1: Create Data Files

Create new tablespaces to store the product library feature classes and tables:

| TABLESPACE     | ArcSDE_PARAMETER          |
|----------------|---------------------------|
| PRODLIB_BDATA  | Business table            |
| PRODLIB_BINDEX | Business table index      |
| PRODLIB_FDATA  | Feature table             |
| PRODLIB_FINDEX | Feature table index       |
| PRODLIB_SDATA  | Spatial Index table       |
| PRODLIB_SINDEX | Spatial Index table index |
| PRODLIB_ADATA  | Adds table (versioned)    |
| PRODLIB_AINDEX | Adds table index          |
| PRODLIB_DDATA  | Deletes table (versioned) |
| PRODLIB_DINDEX | Deletes table index       |

```
ALTER SYSTEM SET WALLET OPEN IDENTIFIED BY "prodlibadmin";
```



J10023

```

CREATE SMALLFILE TABLESPACE "PRODLIB_BDATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Bdata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_BINDE" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Bindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_FDATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Fdata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_FINDEX" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Findex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_SDATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Sdata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_SINDEX" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Sindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_ADATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Adata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_AINDEX" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Aindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_DDATA" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Ddata01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE "PRODLIB_DINDEX" ENCRYPTION default storage (ENCRYPT)
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Dindex01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K SEGMENT SPACE MANAGEMENT AUTO;

```

By setting the data files' initial size to 10 MB, there is no delay in the creation of the tablespaces; to avoid fragmentation, you can resize the data files.

```

ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Bdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Bindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Fdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Findex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Sdata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Sindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Adata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Aindex01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Ddata01.dbf' RESIZE 400M;
ALTER DATABASE DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\prodlib_Dindex01.dbf' RESIZE 400M;

```

## Step 2: Create the PRODLIB User

1. Create a new database user to store the product library feature classes and tables and grant the appropriate permissions:

```

CREATE USER PRODLIB PROFILE DEFAULT IDENTIFIED BY prodlib
DEFAULT TABLESPACE "PRODLIB_BDATA"
TEMPORARY TABLESPACE "TEMP"
QUOTA UNLIMITED ON "PRODLIB_BDATA"
QUOTA UNLIMITED ON "PRODLIB_BINDE"
QUOTA UNLIMITED ON "PRODLIB_FDATA"
QUOTA UNLIMITED ON "PRODLIB_FINDEX"
QUOTA UNLIMITED ON "PRODLIB_SDATA"
QUOTA UNLIMITED ON "PRODLIB_SINDEX"
QUOTA UNLIMITED ON "PRODLIB_ADATA"
QUOTA UNLIMITED ON "PRODLIB_AINDEX"
QUOTA UNLIMITED ON "PRODLIB_DDATA"
QUOTA UNLIMITED ON "PRODLIB_DINDEX"

```

```
ACCOUNT UNLOCK;
```

## 2. Grant privileges:

```
GRANT CREATE SESSION TO PRODLIB;  
GRANT CREATE SEQUENCE TO PRODLIB;  
GRANT CREATE TRIGGER TO PRODLIB;  
GRANT CREATE VIEW TO PRODLIB;  
GRANT CREATE TABLE TO PRODLIB;
```

## Step 3: Modify DBTUNE

### 1. Export the dbtune file before making any modification:

```
sdedbtune -o export -f dbtune_exp.sde -u sde -p sde -i sde:oracle11g:prodlbdb
```

### 2. Copy *dbtune\_exp.sde* to *dbtune\_prodlib.sde*.

### 3. Modify the ##DEFAULTS configuration keywords:

```
dbtune_prodlib.sde  
##DEFAULTS  
A_INDEX_RASTER "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX NOLOGGING"  
A_INDEX_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX STORAGE (INITIAL 409600) NOLOGGING"  
A_INDEX_SHAPE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX STORAGE (INITIAL 409600) NOLOGGING"  
A_INDEX_STATEID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX STORAGE (INITIAL 409600)  
NOLOGGING"  
A_INDEX_USER "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX STORAGE (INITIAL 409600) NOLOGGING"  
A_INDEX_XML "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_AINDEX NOLOGGING"  
A_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_ADATA STORAGE (INITIAL 409600)"  
B_INDEX_RASTER "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX NOLOGGING"  
B_INDEX_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX STORAGE (INITIAL 409600) NOLOGGING"  
B_INDEX_SHAPE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX NOLOGGING"  
B_INDEX_TO_DATE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX NOLOGGING"  
B_INDEX_USER "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX STORAGE (INITIAL 409600) NOLOGGING"  
B_INDEX_XML "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BINDEX NOLOGGING"  
B_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_BDATA STORAGE (INITIAL 409600)"  
D_INDEX_DELETED_AT "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_DINDEX STORAGE (INITIAL 409600)  
NOLOGGING"  
D_INDEX_STATE_ROWID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_DINDEX STORAGE (INITIAL 409600)  
NOLOGGING"  
D_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_DDATA STORAGE (INITIAL 409600)"  
F_INDEX_AREA "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_FINDEX NOLOGGING"  
F_INDEX_FID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_FINDEX NOLOGGING"  
F_INDEX_LEN "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_FINDEX NOLOGGING"  
F_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_FDATA"  
GEOMETRY_STORAGE "ST_GEOMETRY"  
S_INDEX_ALL "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_SINDEX STORAGE (INITIAL 409600) NOLOGGING"  
S_INDEX_SP_FID "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_SINDEX NOLOGGING"  
S_STORAGE "PCTFREE 0 INITRANS 4 TABLESPACE PRODLIB_SDATA STORAGE (INITIAL 409600)"  
ST_GEOM_LOB_STORAGE "STORE AS (TABLESPACE PRODLIB_FDATA ENABLE STORAGE IN ROW CHUNK 8K RETENTION  
CACHE INDEX (TABLESPACE PRODLIB_FINDEX) )"  
END
```

If your database only stores the Product Library repository, you can edit ##DEFAULTS; otherwise, create a new configuration keyword as described above.

### 4. Import the modified *dbtune\_prodlib.sde* file:

```
sdedbtune -o import -f dbtune_prodlib.sde -u sde -p sde -i sde:oracle11g:prodlbdb
```

## Step 4: Create the Product Library Database Connection

Create a database connection in ArcCatalog™ with the PRODLIB user; this will be the product library workspace location.

J10023

## Step 5: Configure Oracle Parameters

It is recommended that you use the following parameter values when creating an Oracle database:

### Oracle Parameters for Product Library

| Parameter Name                                                                                               | Value                                                |
|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
| Configure with Database Enterprise Manager                                                                   | Enabled                                              |
| Automatic Memory Management                                                                                  | Enabled                                              |
| OPEN_CURSORS                                                                                                 | 10000                                                |
| SESSION_CACHED_CURSORS                                                                                       | 50–150                                               |
| DEFERRED_SEGMENT_CREATION                                                                                    | False                                                |
| RESOURCE_LIMIT                                                                                               | True                                                 |
| ALTER PROFILE "DEFAULT" LIMIT IDLE_TIME 60<br>PASSWORD_LIFE_TIME UNLIMITED<br>PASSWORD_GRACE_TIME UNLIMITED; | Set Default profile for ArcSDE and data owner users. |
| RECYCLEBIN                                                                                                   | Off                                                  |

## Step 6: Configure ArcSDE Parameters

You need to configure the MAXBLOBSIZE and TCPKEEPALIVE parameters for the ArcSDE geodatabase used as the product library. The MAXBLOBSIZE value is -1 by default. However, if you are using Oracle or another enterprise DBMS, make sure that this value is set to -1 and the TCPKEEPALIVE value is set to 1. This command should be used from the command prompt of a machine where ArcSDE is installed.

```
sdeconfig -o alter -v MAXBLOBSIZE=-1 -i <service> -u sde -p <sde_password>
sdeconfig -o alter -v TCPKEEPALIVE=1 -i <service> -u sde -p <sde_password>
```

For more information, see the ArcSDE Administration Command Reference.

## Step 7: Create the CKB\_USERS Role

A role needs to be assigned to the users who are going to be working with the product library so they can view or add components, information, and data. The role CKB\_USERS must be created for the users to be recognized by the product library. This can be done using the following statement:

```
CREATE ROLE CKB_USERS NOT IDENTIFIED
```

## Step 8: Create the Product Library Workspace

After the geodatabase has been created, various tables and feature classes that are part of the product library need to be added to it. This process can be completed in ArcMap™.

The steps in this section are for defining and upgrading the geodatabase as a product library in ArcMap.

### Steps

1. Start ArcMap.

2. On the menu bar, click **Customize > Production > Product Library**.

**Tips:**

- If the **Product Library** command is not enabled, you may need to enable the Production Mapping extension by clicking **Customize > Extensions** and checking the check box for **Production Mapping**.
- You can also open the window by clicking the **Product Library** button on the **Production Cartography** or **Production Editing** toolbar.

3. Right-click **Product Library** and click **Select Product Library**.

The **Choose Product Library Workspace** dialog box appears.

4. Navigate to the product library database.
5. Click **Open**.

The **Upgrade Product Library Workspace** or the **Create Product Library Workspace** dialog box appears if the geodatabase does not have all the components necessary to perform as a product library. Continue to step 6 if one of these dialog boxes appears.

**Note:** If you also have Esri Nautical Solution installed, tables are added to the product library schema that are necessary for nautical production but do not impact nonnautical production.

If this is not a new product library, and if an error appears when you choose a geodatabase as the product library, it may need to be compacted or compressed to restore its integrity. Compacting applies to personal and file geodatabases, and compressing applies to enterprise geodatabases.

If you select an enterprise workspace, the **Select Product Library Owner** dialog box appears, and you can continue to step 7.

6. Perform the following steps to upgrade or create the geodatabase you want to use as the product library:
  - a. If necessary, click the drop-down arrow in the **Select Configuration Keyword** area and choose an option.

Options vary based on whether the database is a personal, file, or ArcSDE geodatabase.
  - b. Click **OK**.
  - c. Click **OK** once the upgrade or creation completes.
7. If the product library is an enterprise geodatabase, click the owner and click **OK**.

The **Upgrade Product Library Workspace** or the **Create Product Library Workspace** dialog box appears if the geodatabase does not have all the components necessary to perform as a product library. Perform the following steps if you are an

administrator and want to upgrade or create the geodatabase to use as the product library:

- a. If necessary, click the drop-down arrow in the **Select Configuration Keyword** area and choose an option.

Options vary based on whether the database is a personal, file, or ArcSDE geodatabase.

- b. Click **OK**.
- c. If necessary, click **OK** once the upgrade or creation completes.

## Step 9: Verify the Storage

Run the SQL queries below to verify that the product library workspace was created under the correct tablespaces:

```
sqlplus prodlib/prodlib@prodlibdb
--TABLES--
SELECT TABLE_NAME, TABLESPACE_NAME, STATUS FROM USER_TABLES ORDER BY 1;
--INDEXES--
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS FROM USER_INDEXES ORDER BY 2,1;
--LOBS--
SELECT * FROM USER_LOBS WHERE SEGMENT_NAME LIKE 'SYS_LOB%' ORDER BY TABLE_NAME;
```

If any tables or indexes are stored in the wrong tablespace, use **ALTER TABLE** and **ALTER INDEX** to change the tablespace. See the SQL syntax.

```
ALTER TABLE <table_name> MOVE TABLESPACE <tablespace_name>;
ALTER INDEX <index_name> REBUILD TABLESPACE <tablespace_name>;
```

If moving large objects (LOB), read Oracle MetaLink Doc ID: 130814.1, "How to Move LOB Data to Another Tablespace."

## Step 10: Register as Versioned

If you manually import the product library tables and feature classes, such as importing an XML file, you need to verify that all the tables and feature classes are registered as versioned. This allows the software to edit the tables as you create and work with your product library. However, you shouldn't create new versions of the product library tables; this can result in inconsistencies within the versions.

### Steps

1. Expand **Database Connections** in the **Catalog Tree** window.
2. Double-click the product library administrator connection geodatabase to connect to it.
3. Right-click each feature class in your product library and click **Manage > Register as Versioned**.

Do not check the **Register the selected objects with the option to move edits to base** check box.

4. Click **OK**.

5. Right-click each table in your product library, except for those listed below, and click **Manage > Register as Versioned**.

Do not check the **Register the selected objects with the option to move edits to base** check box.

Do not register the following tables for versioning:

- ELM\_CATEGORIES
- ELM\_ELEMENTS
- ELM\_PRODUCTS
- ELM\_SOLUTIONS

6. Click **OK**.

## Step 11: Validate Permissions and Roles

All the tables in the product library need to have read/write privileges assigned to them except the PCAT\_PERMISSION table. The PCAT\_PERMISSION table only needs read privileges assigned to it. The permissions need to be assigned to the CKB\_USERS role. You can re-create the CKB\_USERS role and grant the right permissions to the role by using the following script:

```
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;
SET SERVEROUTPUT ON;
spool ROLE_CKB_USERS.sql;

select 'DROP ROLE "CKB_USERS";' from dual;
select 'CREATE ROLE "CKB_USERS" NOT IDENTIFIED;' from dual;

select 'grant select,insert,update,delete on ' || owner || '.' || table_name || ' to CKB_USERS;'
from sys.dba_tables where lower(owner) = 'prodlib' order by table_name;

select 'REVOKE INSERT,UPDATE,DELETE ON PRODLIB.PCAT_PERMISSION FROM CKB_USERS;' from dual;

spool off;
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;

SET SERVEROUTPUT ON;
/
@ROLE_CKB_USERS.sql;
/
```

## Grant Permissions Using ArcCatalog

Both the administrator and other user accounts in the underlying database management system should have appropriate privileges and roles assigned to them. When you set up your connection to your spatial database, ensure that you are connecting as the appropriate user.

## Steps

1. Start ArcCatalog.
2. Expand **Database Connections** in the **Catalog Tree** window.
3. Double-click the product library administrator connection geodatabase to connect to it.
4. Select all tables except PCAT\_PERMISSION, right-click, then click **Manage > Privileges**.
5. Type CKB\_USERS into the **User** text box on the **Change Privileges** dialog box.
6. Click the **View (Select)** drop-down arrow and choose **GRANT**.
7. Click the **Edit (Update/Insert/Delete)** drop-down arrow and choose **GRANT**.
8. Click **OK**.
9. Select the **PCAT\_PERMISSION** table, right-click, then click **Privileges**.
10. Type CKB\_USERS into the **User** text box on the **Change Privileges** dialog box.
11. Click the **View** drop-down arrow and choose **GRANT**.
12. Click **OK**.

## Step 12: Configure Log File Tables

Enterprise geodatabases use log file tables to maintain lists of selected records. Records are written to log file tables for later use by the application whenever a selection of a specific size is made, a reconciliation or post on a versioned database is performed, or a disconnected editing checkout is done in a client application. The log file tables store the ObjectIDs of the selected features so they can be redisplayed. This allows faster analysis and processing of information.

In ArcGIS® software, by default, log file tables are used if the selection set contains 100 or more records. This selection threshold of 100 features is set in the registry. It can be changed; however, Esri does not recommend doing so. There is no proven performance reason for changing it, and doing so could cause performance problems. Thus, log file tables store feature selections in ArcMap that are greater than 100 for each connected ArcSDE editor/viewer user. Hence, it is recommended that you store the log file tables in a separate tablespace; this can be achieved with the DBTUNE table.

Log file options are set using specific parameters in the SERVER\_CONFIG and DBTUNE tables. Parameters in these tables are altered using the sdeconfig and sdedbtune commands, respectively.

## Create Log File Tablespaces

```
CREATE SMALLFILE TABLESPACE SDELOGFILE
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\SDE\sdelogfile01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE
800M LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K SEGMENT SPACE MANAGEMENT AUTO;

CREATE SMALLFILE TABLESPACE SDELOGFILEIDX
DATAFILE 'D:\oracle\ORADATA\PRODLIBDB\SDE\sdelogfileidx01.dbf' SIZE 10M AUTOEXTEND ON NEXT 1M
MAXSIZE 400M LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K SEGMENT SPACE MANAGEMENT AUTO;
```

## ***Change DBTUNE Log File Parameters***

1. Export the DBTUNE table:

```
sdedbtune -o export -f dbtune_logfile.sde -u sde -p sde -i sde:oracle11g:prodlbdb
```

2. Modify the dbtune\_logfile.sde ##LOGFILE\_DEFAULTS configuration keyword:

```
##LOGFILE_DEFAULTS
LD_INDEX_DATA_ID          "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
LF_INDEXES                 "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
LF_STORAGE                 "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE"
SESSION_INDEX             "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
SESSION_TEMP_TABLE        0
SESSION_STORAGE           "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE"
LD_STORAGE                "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILE "
LD_INDEX_ROWID            "PCTFREE 0 INITRANS 4 TABLESPACE SDELOGFILEIDX NOLOGGING "
END
```

3. Import the modified dbtune\_logfile.sde:

```
sdedbtune -o export -f dbtune_logfile.sde -u sde -p sde -i sde:oracle11g:mcs
```

## ***Create Log File Tables***

1. Grant QUOTA permissions to the users on SDELOGFILE and SDELOGFILEIDX:

```
ALTER USER SDE QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER SDE QUOTA UNLIMITED ON "SDELOGFILEIDX";

ALTER USER PRODLIB QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER PRODLIB QUOTA UNLIMITED ON "SDELOGFILEIDX";

ALTER USER PRODLIBUSER QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER PRODLIBUSER QUOTA UNLIMITED ON "SDELOGFILEIDX";
```

2. Grant CREATE TABLE permission for the ArcSDE editor/viewer user.
3. In ArcMap, select more than 100 features; this automatically creates the log file tables.
4. Remove CREATE TABLE permissions as appropriate.

Learn more about ArcSDE log file tables at

[resources.arcgis.com/en/help/main/10.1/index.html#//002n00000014000000](http://resources.arcgis.com/en/help/main/10.1/index.html#//002n00000014000000).

## **Step 13: Create the ArcSDE Product Library User**

The example below shows how to create an ArcSDE user to access the product library:

1. Create the PRODLIBUSER user:

```
CREATE USER PRODLIBUSER PROFILE "DEFAULT"
IDENTIFIED BY mapeditor
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
```

2. Grant privileges:

```
GRANT "CONNECT" TO "PRODLIBUSER";
GRANT "CREATE TABLE" TO "PRODLIBUSER";
GRANT "CKB_USERS" TO "PRODLIBUSER";
```

3. Grant QUOTA on log file tablespaces:

```
ALTER USER PRODLIBUSER QUOTA UNLIMITED ON "SDELOGFILE";
ALTER USER PRODLIBUSER QUOTA UNLIMITED ON "SDELOGFILEIDX";
```



J10023

### **Step 14: Create Database Connections for Product Library Users**

Database connections need to be created for the other product library users if the product library is stored in Oracle. Create a database connection in ArcCatalog with the PRODLIBUSER user; this will be the product library ArcSDE connection. **Note:** If you are using database authentication, type the user name of the product library user for which you want to create a database connection.

### **Step 15: Assign Product Library Permissions**

There are two different levels of interaction with the product library in a Spatial Database Engine™ (SDE™) implementation of the product library: administrators and users. These levels of access are controlled through ArcCatalog database connections. The administrators manage the overall product library including the structure, components, and user permissions. This level of permissions through ArcCatalog database connections is related to the database role CKB\_USERS.

Users can have varying degrees of access to parts of the product library determined by whether they have edit, read/write, or read-only permissions based on their Windows login. Using the administrator's database connection, user accounts are created for anyone who is going to have access to the product library. To create a new user, you must first add the person as a product library user, then assign permissions.

Learn more about product library permissions:

[resources.arcgis.com/en/help/main/10.1/index.html#/Product\\_library\\_permissions/010300000043000000/](https://resources.arcgis.com/en/help/main/10.1/index.html#/Product_library_permissions/010300000043000000/)

### **Step 16: Add New Users to the Product Library**

Using the administrator's database connection, user accounts are created for anyone who is going to have access to the product library. To create a new user, the person must first be added as a product library user, then permissions can be assigned.

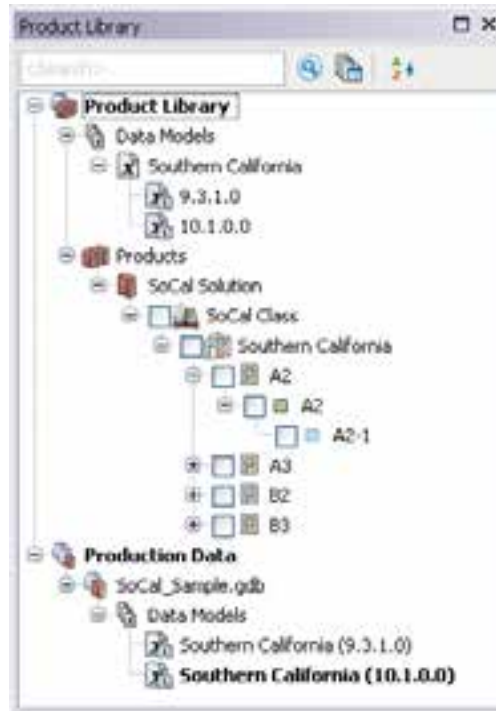
Initially, the user must be added to the geodatabase by the administrator. Each user is added using the first name, last name, and Windows user name.

**Note:** This only applies if you are using an ArcSDE geodatabase as your product library. Personal and file geodatabase permissions are defined by the user's permissions at the operating system level.

#### **Steps**

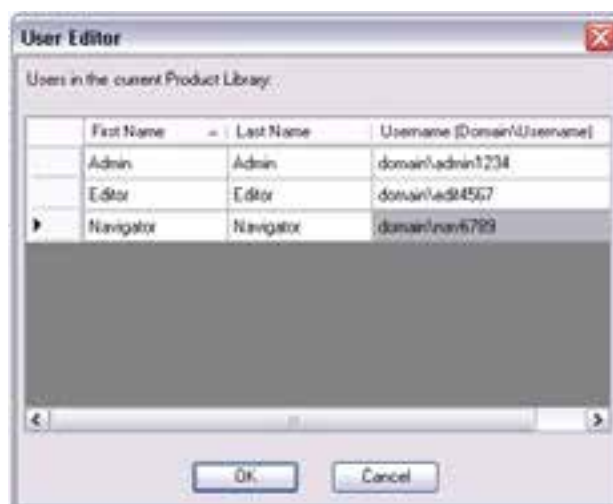
1. Start ArcMap.
2. If necessary, open the **Product Library** window by clicking **Customize > Production > Product Library** on the main menu.

A tree view of the product library appears.



3. If necessary, define the product library workspace.
4. Right-click **Product Library** and click **Configure > Users**.

The **User Editor** dialog box appears.



**Tip:** If you are using an ArcSDE geodatabase as your product library, you can also add new users to the product library by right-clicking a series and clicking **Permissions**.

5. Right-click anywhere in the **Users in the current Product Library** list and click **New User**.

A new row appears in the list.

**Tip:** If you are using an ArcSDE geodatabase as your product library, you can also add users when you are assigning permissions to existing users.

6. Type the user's first name in the **First Name** cell.
7. Type the user's last name in the **Last Name** cell.
8. Type the user's Windows login name in the **Username (Domain\Username)** cell.
9. Repeat steps 5 through 8 for each user you need to add to the product library.
10. Click **OK**.

If the user name(s) is valid, the user(s) is added to the product library.

### ***Assigning Permissions to Users***

Once the user is added, the permissions can be granted at the series level of the product library. Permissions are passed down to all products within a given series. Permissions are also passed up from the series to the class and the solution. For example, if a user is given permissions to one or more series below a particular class or solution, the user has access to those entries. By default, the permissions for a user are set to Not Available, but there are four different levels:

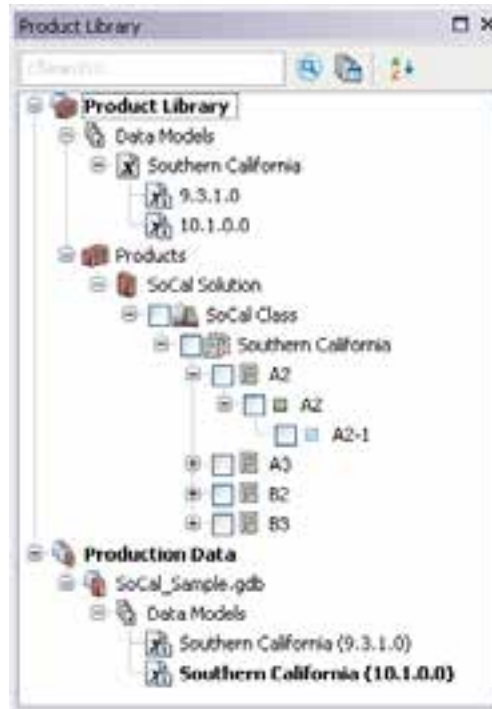
- Not Available—The series and all components beneath it are hidden from the user.
- Read Only—Properties can be viewed for all levels of the product library.
- Check In/Check Out—Files can be checked in and out.
- Edit—Product library levels can be added, modified, and removed, and files can be checked in and out.

**Note:** This only applies if you are using an ArcSDE geodatabase as your product library. Personal and file geodatabase permissions are defined by the user's permissions at the operating system level.

#### **Steps**

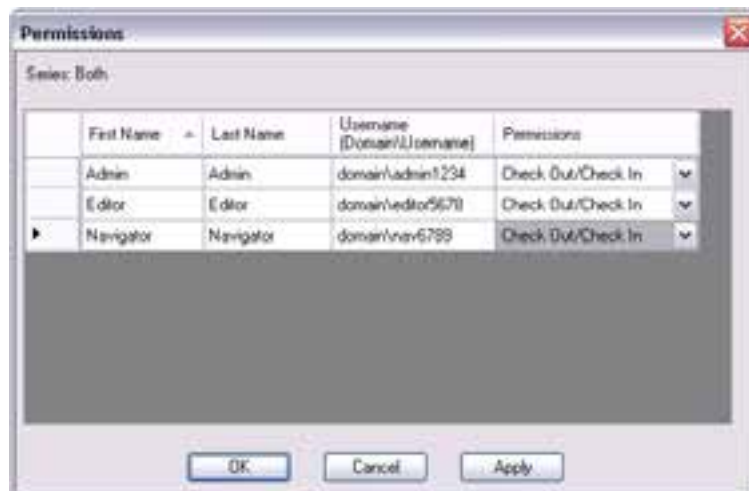
1. Start ArcMap.
2. If necessary, open the **Product Library** window by clicking **Customize > Production > Product Library** on the main menu.

A tree view of the product library appears.



3. If necessary, expand **Product Library** and **Products**.
4. Navigate to the series level of the product class for which you want to assign permissions.
5. Right-click the series name and click **Permissions**.

The **Permissions** dialog box appears.



6. Click the **Permissions** drop-down arrow for the user to whom you want to grant permissions and choose an option.
7. Repeat step 6 for all users to whom you want to give permissions.
8. Click **OK**.

### **Replication**

You can use geodatabase replication to replicate the Product Library workspace (one-way, two-way replication), but you can create a *new* solution, product class, series, or product only in the parent geodatabase *or* only in the child geodatabase.

### **Conclusion**

You can reduce disk contention and improve database I/O by storing the product library workspace in different locations on disk. However, this practice alone does not guarantee optimal database performance, and additional tuning tasks may be needed.

Learn more about the recommended tuning tasks:

[resources.arcgis.com/en/help/main/10.1/index.html#/Minimize\\_disk\\_I\\_O\\_contention\\_in\\_Oracle/002n00000025000000/](https://resources.arcgis.com/en/help/main/10.1/index.html#/Minimize_disk_I_O_contention_in_Oracle/002n00000025000000/)

For more information on the product library, visit the Esri Production Mapping page: [esri.com/software/arcgis/extensions/production-mapping/index.html](https://esri.com/software/arcgis/extensions/production-mapping/index.html)

Learn about setting up the product library in an ArcSDE environment:

[resources.arcgis.com/en/help/main/10.1/index.html#/in\\_an\\_enterprise\\_geodatabase/0103000002sn000000/](https://resources.arcgis.com/en/help/main/10.1/index.html#/in_an_enterprise_geodatabase/0103000002sn000000/)

Access blogs, forums, downloads, and more, via the Esri Production Mapping resource center:

[resources.arcgis.com/en/communities/production-mapping/](https://resources.arcgis.com/en/communities/production-mapping/)

You can access other resources at ArcGIS 10.1 for Desktop Help:

[resources.arcgis.com/en/help/main/10.1/#/Welcome\\_to\\_the\\_ArcGIS\\_Professional\\_Help\\_Library/00qn0000001p000000/](https://resources.arcgis.com/en/help/main/10.1/#/Welcome_to_the_ArcGIS_Professional_Help_Library/00qn0000001p000000/) and Esri Support ([support.esri.com](https://support.esri.com))



Esri inspires and enables people to positively impact their future through a deeper, geographic understanding of the changing world around them.

Governments, industry leaders, academics, and nongovernmental organizations trust us to connect them with the analytic knowledge they need to make the critical decisions that shape the planet. For more than 40 years, Esri has cultivated collaborative relationships with partners who share our commitment to solving earth's most pressing challenges with geographic expertise and rational resolve. Today, we believe that geography is at the heart of a more resilient and sustainable future. Creating responsible products and solutions drives our passion for improving quality of life everywhere.



## Contact Esri

380 New York Street  
Redlands, California 92373-8100 USA

1 800 447 9778  
T 909 793 2853  
F 909 793 5953  
info@esri.com  
[esri.com](http://esri.com)

Offices worldwide  
[esri.com/locations](http://esri.com/locations)